Course Title: Artificial Intelligence **Sheet #5**

The file ex3data1.txt contains the dataset for our sheet.

1. Implement the K-means clustering algorithm and apply it to the given 2D dataset that will help you gain an intuition of how the K-means algorithm works. Implement the two phases of the K-means algorithm separately.

   In the "cluster assignment" phase of the K-means algorithm, the algorithm assigns every training example $x^{(i)}$ to its closest centroid, given the current positions of centroids. Given assignments of every point to a centroid, the second phase of the algorithm recomputes, for each centroid, the mean of the points that were assigned to it.

a) Complete the code in findClosestCentroids.m. This function takes the data matrix X and the locations of all centroids inside centroids and should output a one-dimensional array idx that holds the index (a value in {1,.....,K}, where K is total number of centroids) of the closest centroid to every training example.
b) Complete the code in computeCentroids.m. for the second phase of the algorithm.
c) After completing the two functions computeCentroids and findClosestCentroids, you have all the necessary pieces to write the k-Means algorithm.
d) Write your main function that run the k-means algorithm. First load the data and display it on a 2-dimensional plot. Run K-means algorithm and display your progress.

Best wishes

Dr. Sherin El Gokhy

```
function centroids = computeCentroids(X, idx, K)
%COMPUTECENTROIDS returs the new centroids by computing the means of the
%data points assigned to each centroid.
%   centroids = COMPUTECENTROIDS(X, idx, K) returns the new centroids by
%   computing the means of the data points assigned to each centroid. It is
%   given a dataset X where each row is a single data point, a vector
%   idx of centroid assignments (i.e. each entry in range [1..K]) for each
%   example, and K, the number of centroids. You should return a matrix
%   centroids, where each row of centroids is the mean of the data points
%   assigned to it.
%

% Useful variables
[m n] = size(X);

% You need to return the following variables correctly.
centroids = zeros(K, n);


% ====================== YOUR CODE HERE ======================
% Instructions: Go over every centroid and compute mean of all points that
%               belong to it. Concretely, the row vector centroids(i, :)
%               should contain the mean of the data points assigned to
%               centroid i.
%
% Note: You can use a for-loop over the centroids to compute this.




% =============================================================


end
```

```
function idx = findClosestCentroids(X, centroids)
%FINDCLOSESTCENTROIDS computes the centroid memberships for every example
%   idx = FINDCLOSESTCENTROIDS (X, centroids) returns the closest centroids
%   in idx for a dataset X where each row is a single example. idx = m x 1
%   vector of centroid assignments (i.e. each entry in range [1..K])
%

% Set K
K = size(centroids, 1);

% You need to return the following variables correctly.
idx = zeros(size(X,1), 1);

% ====================== YOUR CODE HERE ======================
% Instructions: Go over every example, find its closest centroid, and store
%               the index inside idx at the appropriate location.
%               Concretely, idx(i) should contain the index of the centroid
%               closest to example i. Hence, it should be a value in the
%               range 1..K
%
% Note: You can use a for-loop over the examples to compute this.



% =============================================================

end
```